"Run to the Hills" (Maiden, I.).

Hazards de Controle

Paulo Ricardo Lisboa de Almeida

DSBD

UFPR
UNIVERSIDADE FEDERAL DO PARANÁ

Hazards de Controle

Por que as instruções a seguir geram um hazard de controle?

```
endereço      instrução

40            beq $1,$3,7
44            and $12,$2,$5
48            or $13,$6,$2
52            add $14,$2,$2
…             …
72            lw $4,50($7)
…             …
```

Hazards de Controle

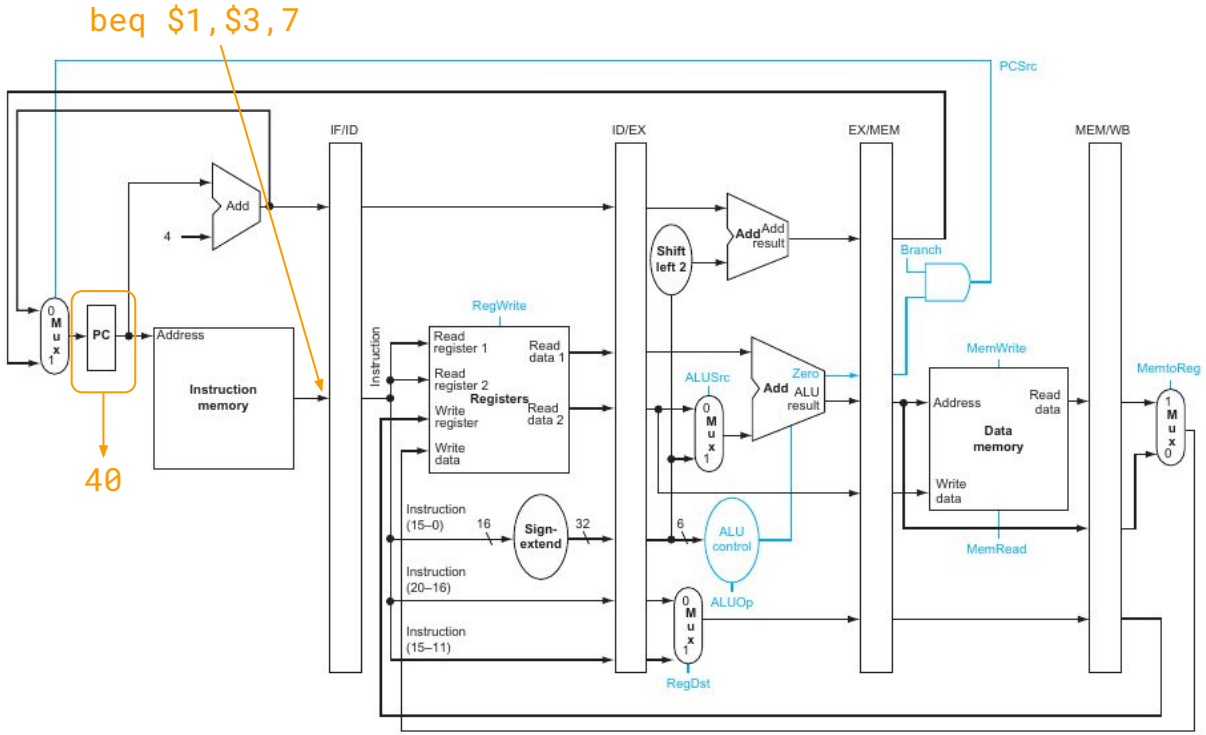
Por que as instruções a seguir geram um hazard de controle?

```
endereço      instrução

40            beq $1,$3,7 #Não sabemos qual será a próxima instrução
44            and $12,$2,$5 #Essa?
48            or $13,$6,$2
52            add $14,$2,$2
…             …
72            lw $4,50($7) #Ou essa?
…             …
```

Exemplo

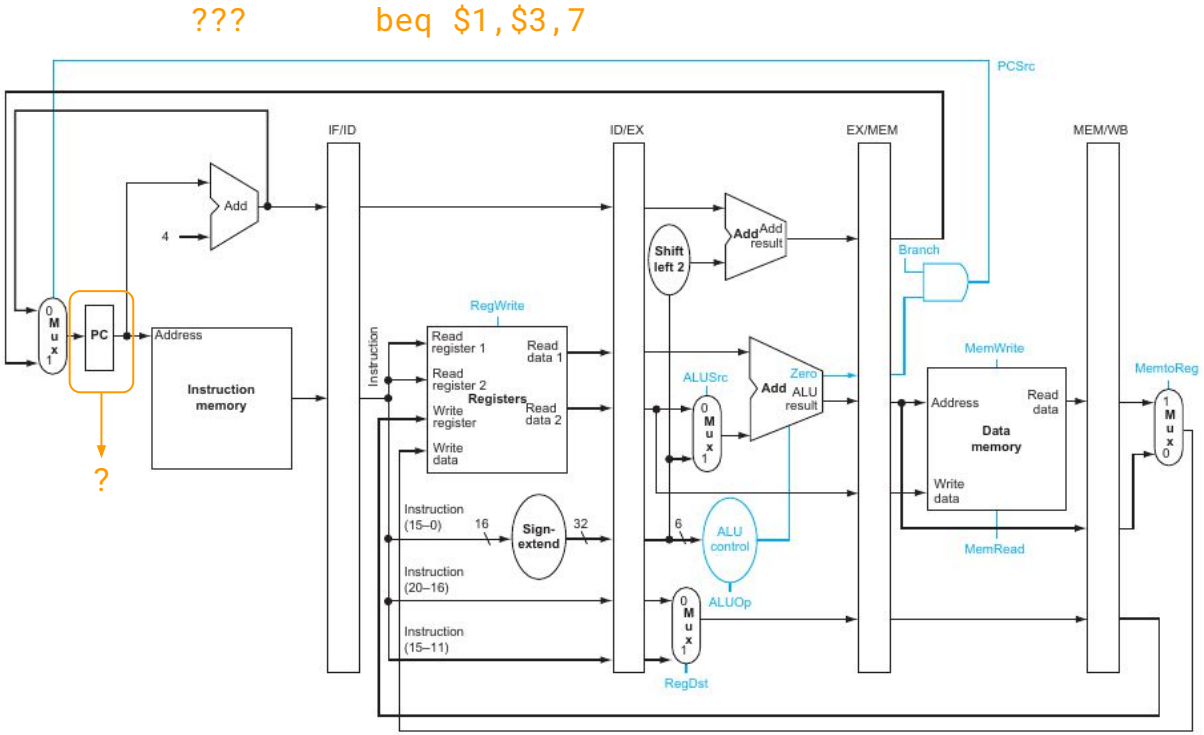
endereço	instrução
40	beq $1,$3,7
44	and $12,$2,$5
48	or $13,$6,$2
52	add $14,$2,$2
…	…
72	lw $4,50($7)
…	…



Exemplo

O beq ainda não terminou de ser executado, e não sabemos se devemos executar o salto ou não.

endereço	instrução
40	beq $1,$3,7
44	and $12,$2,$5
48	or $13,$6,$2
52	add $14,$2,$2
…	…
72	lw $4,50($7)
…	…



Hazards de Controle

Podemos fazer um **pipeline stall**.

- Inserir bolhas (nops).
- Ineficiente.

Outras soluções?

Hazards de Controle

Podemos fazer um **pipeline stall**.

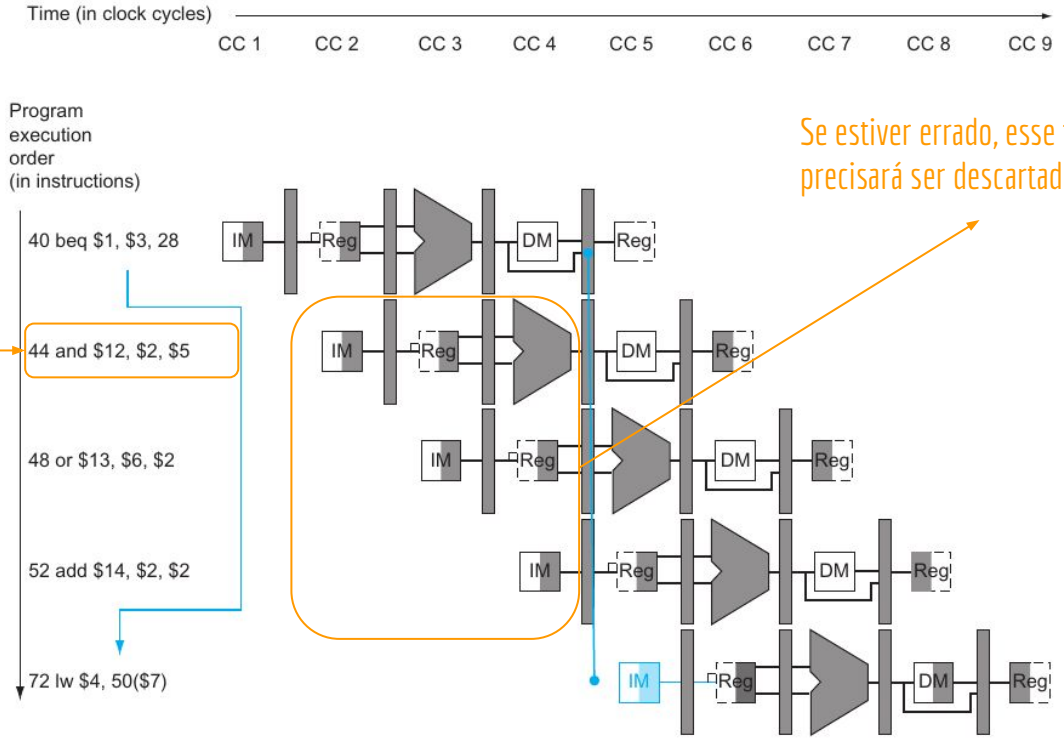
- Inserir bolhas (nops).
- Ineficiente.

Vamos partir de uma solução simples:

Partir do princípio que o **desvio nunca é tomado** (sempre carregar e iniciar a próxima instrução).

- Se estivermos errados, desfazemos tudo e continuamos a partir do endereço correto.
- Podemos assumir que acertamos cerca de 50% das vezes.

Assumir que o desvio nunca é tomado

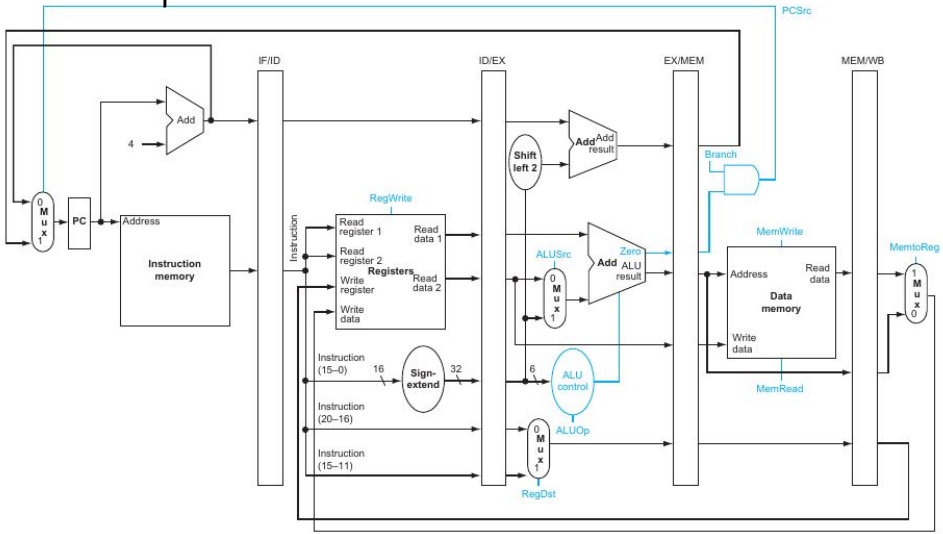


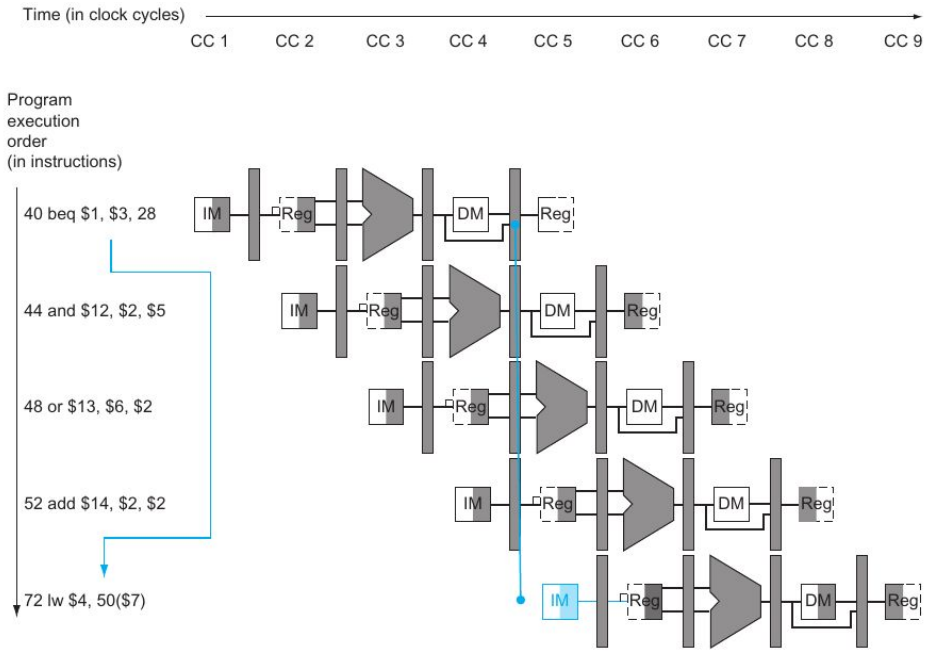
Hazards de Controle

Caso a previsão esteja incorreta, 3 instruções precisam ser descartadas.

Uma no estágio EX, uma no ID, e uma no IF.

Como fazer esse descarte?



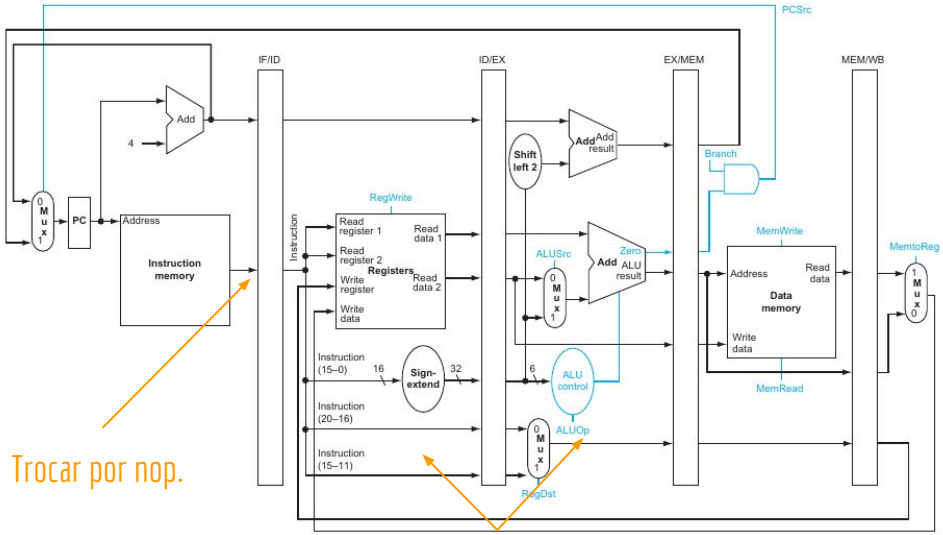


Hazards de Controle

Caso a previsão esteja incorreta, 3 instruções precisam ser descartadas.

Uma no estágio EX, uma no ID, e uma no IF.

Como fazer esse descarte?



Trocar por nop.

Zerar sinais de controle para instruções nesses estágios.

De maneira similar à detecção de hazards de dados, colocar zeros nos sinais de controle dessas instruções para que nada seja alterado.

Se tornam nops.

Precisamos ainda injetar um nop no lugar da instrução que está no primeiro estágio.

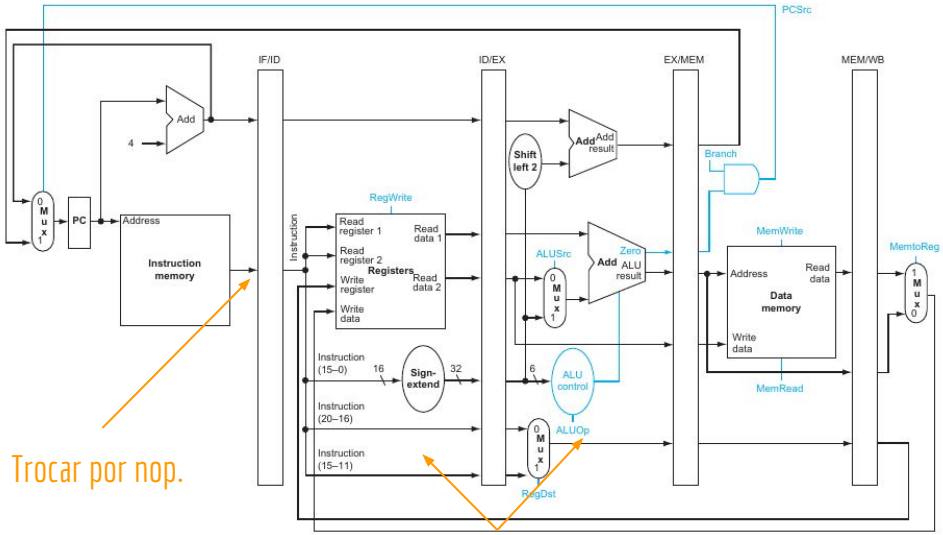
Trocar seu opcode.

Nops

No caso de “previsão incorreta”, estamos descartando três instruções.

Três nops são processados.

Três ciclos de clock inutilizados.



Trocar por nop.

Zerar sinais de controle para instruções nesses estágios.

Reduzindo atrasos

No processador MIPS visto na disciplina.

Por enquanto assumimos que o resultado do branch só está pronto a partir dos registradores EX/MEM.

A instrução deve estar no estágio MEM.

Se conseguirmos calcular os resultados antes, podemos reduzir o custo de uma previsão incorreta.

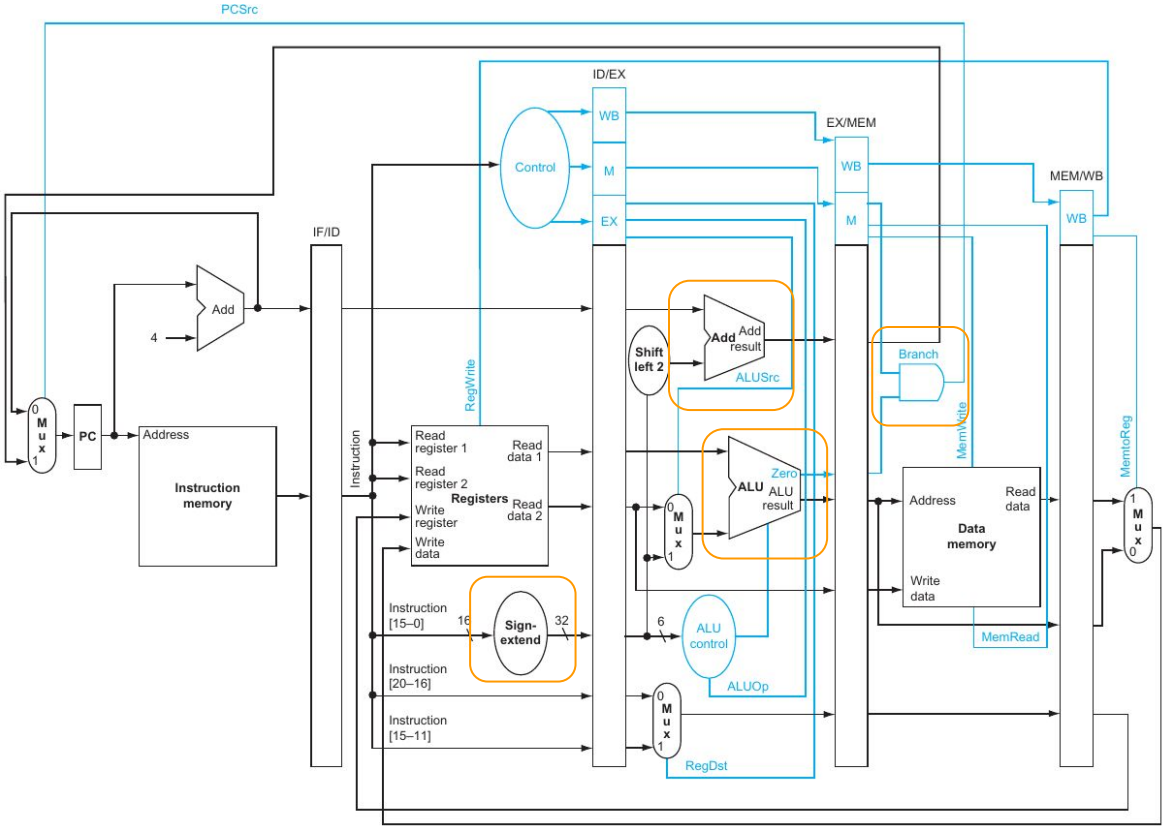
Chegamos a conclusão se o desvio está incorreto antes.

Se a previsão está incorreta, menos trabalho executado é jogado fora.

Reduzindo atrasos

Em destaque estão algumas das principais estruturas envolvidas em um branch.

Como realizar tudo no estágio ID?



Reduzindo atrasos

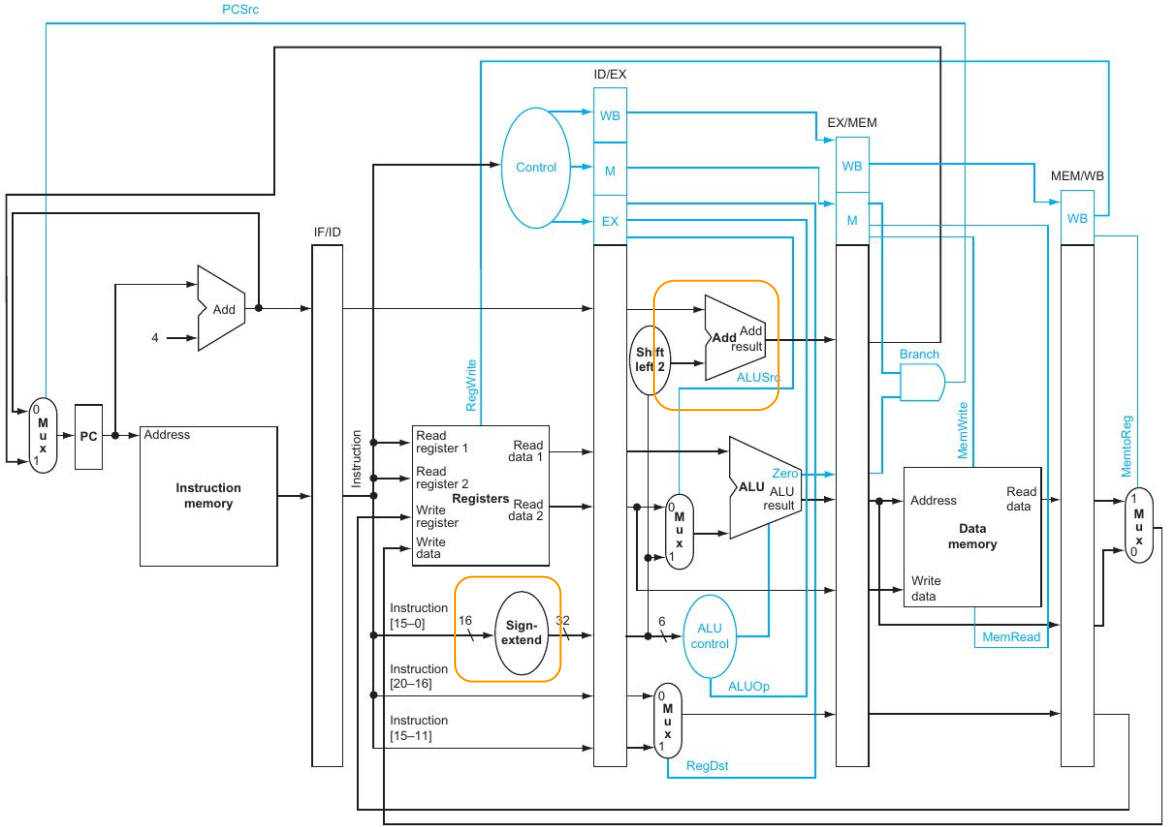
Calcular o endereço do salto deslocando os bits do imediato para a esquerda 2x e somando com PC + 4.

Informações já disponíveis no estágio ID.

Mudança simples, basta mover os componentes responsáveis.

Não há atraso extra.

Esses valores podem ser calculados em paralelo, enquanto o banco de registradores busca pelas informações e a unidade de controle gera os sinais.



Reduzindo atrasos

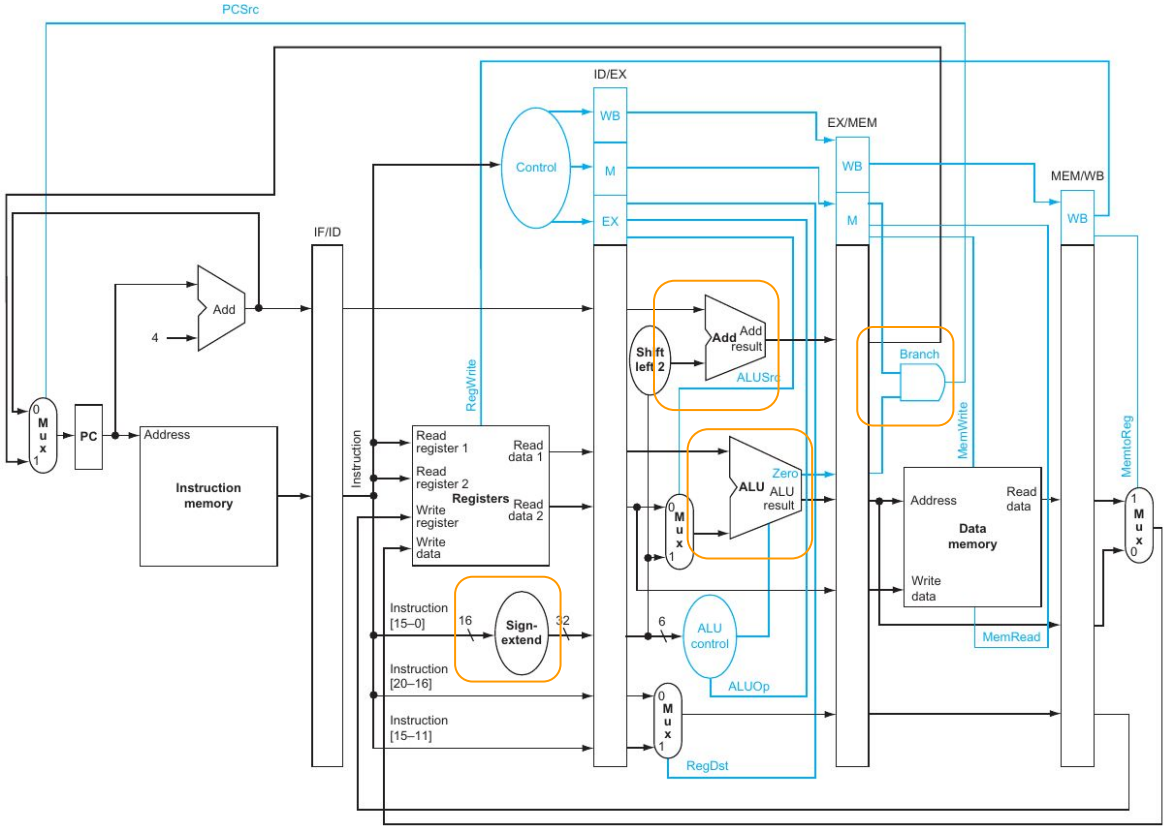
A **comparação é mais complicada.** Quem está fazendo é a ALU através de uma subtração.

Podemos criar um circuito rápido especialista, que faz a comparação.

Um **xnor** bit a bit entre duas palavras retorna 1 se elas forem iguais.

REG1 xnor REG2.

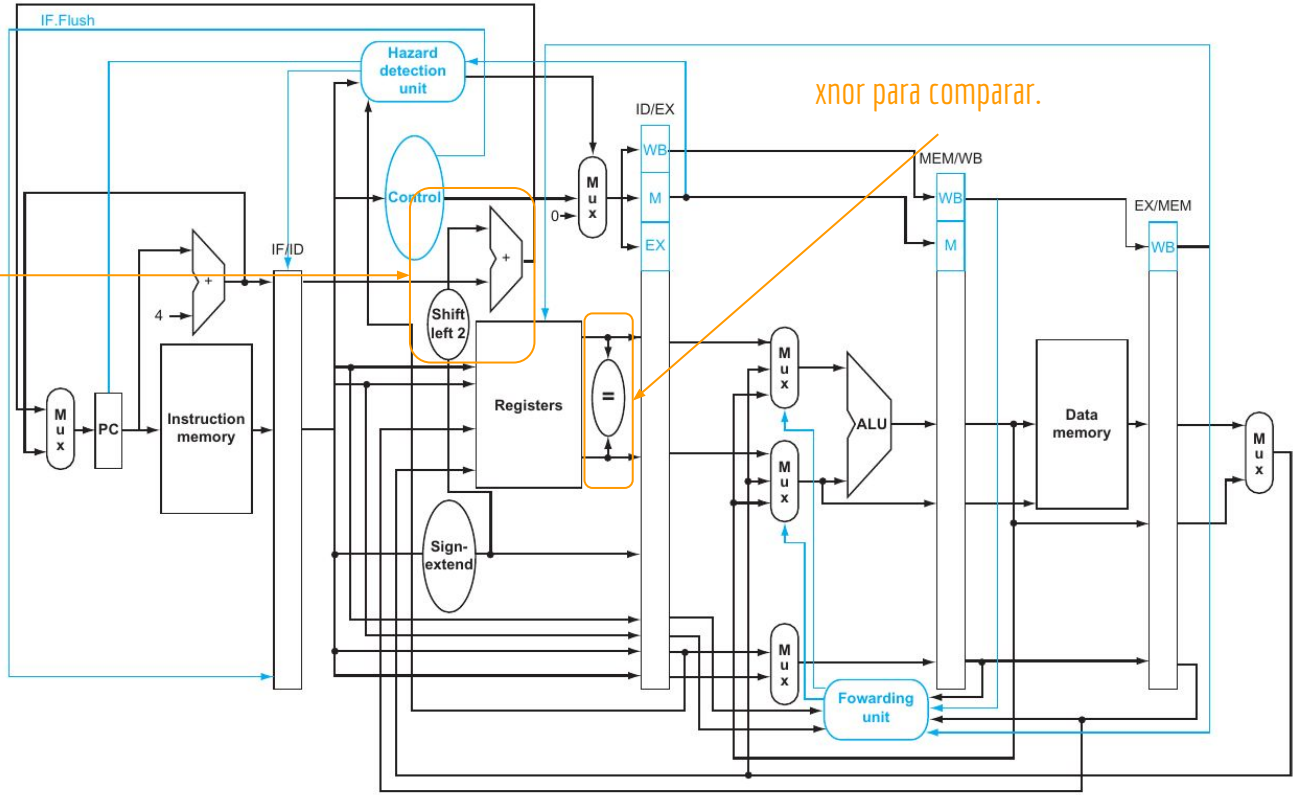
Precisamos colocar esse circuito após a carga dos dados dos registradores.



Realizar a comparação no estágio ID

Cálculo do endereço.

Nessa versão estilizada estão ocultas boa parte dos sinais para simplificar. Ex.: sinal enviado pelo comparador de registradores.

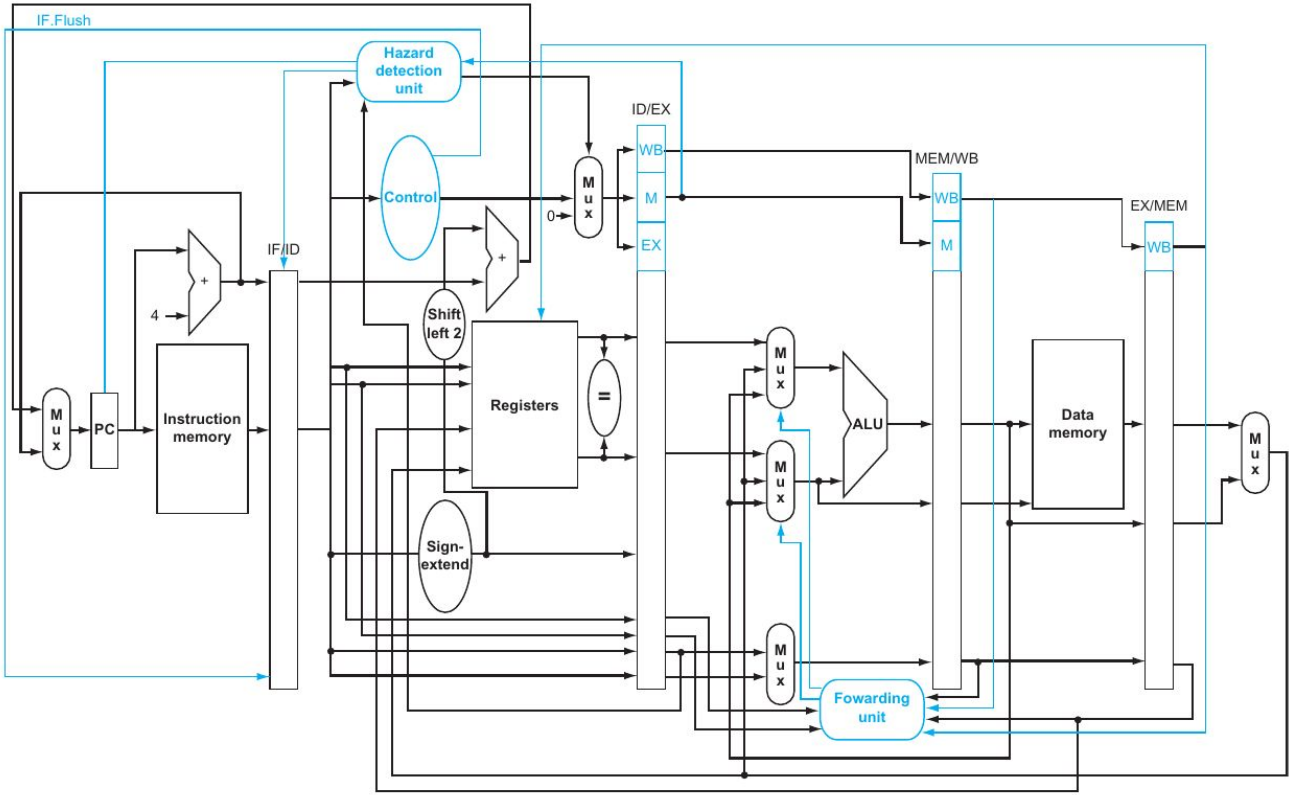


Faça você mesmo

Quais são os custos envolvidos nesta alteração?

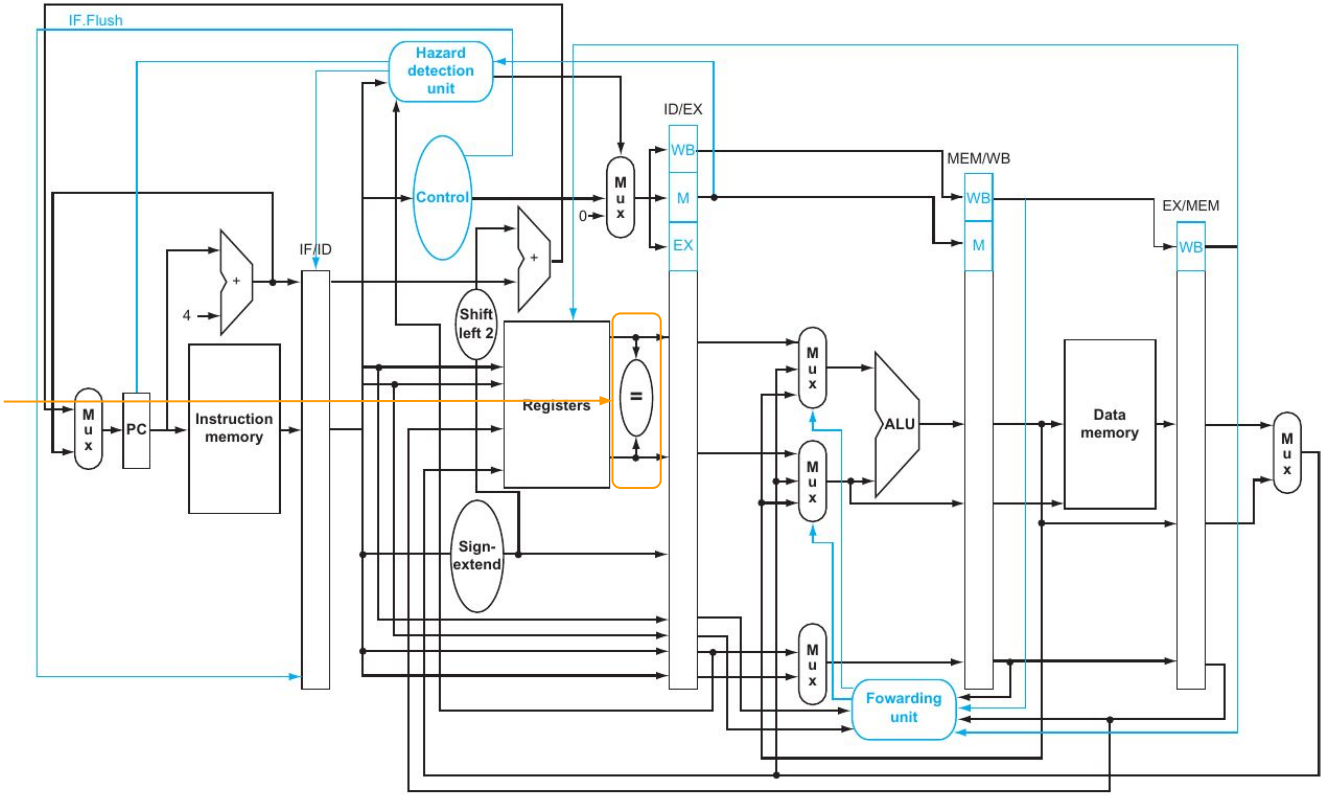
R$, tempo, complexidade, ...

Trazer a comparação para um estágio anterior adicionando mais hardware como feito é sempre uma boa ideia?



Realizar a comparação no estágio ID

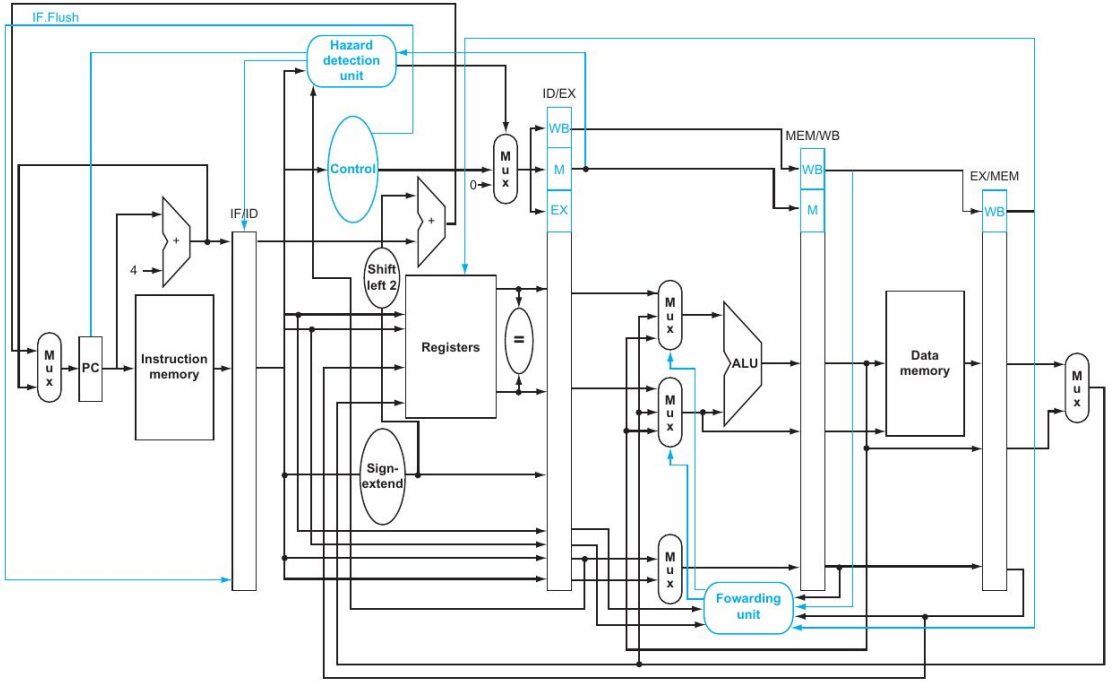
Pode ser necessário um tempo extra nesse estágio devido a comparação, que só pode ser feita após a carga dos registradores.



Realizar a comparação no estágio ID

Redução do custo de uma previsão incorreta.

Agora no máximo uma bolha (nop) é necessária devido ao hazard de controle.



Realizar a comparação no estágio ID

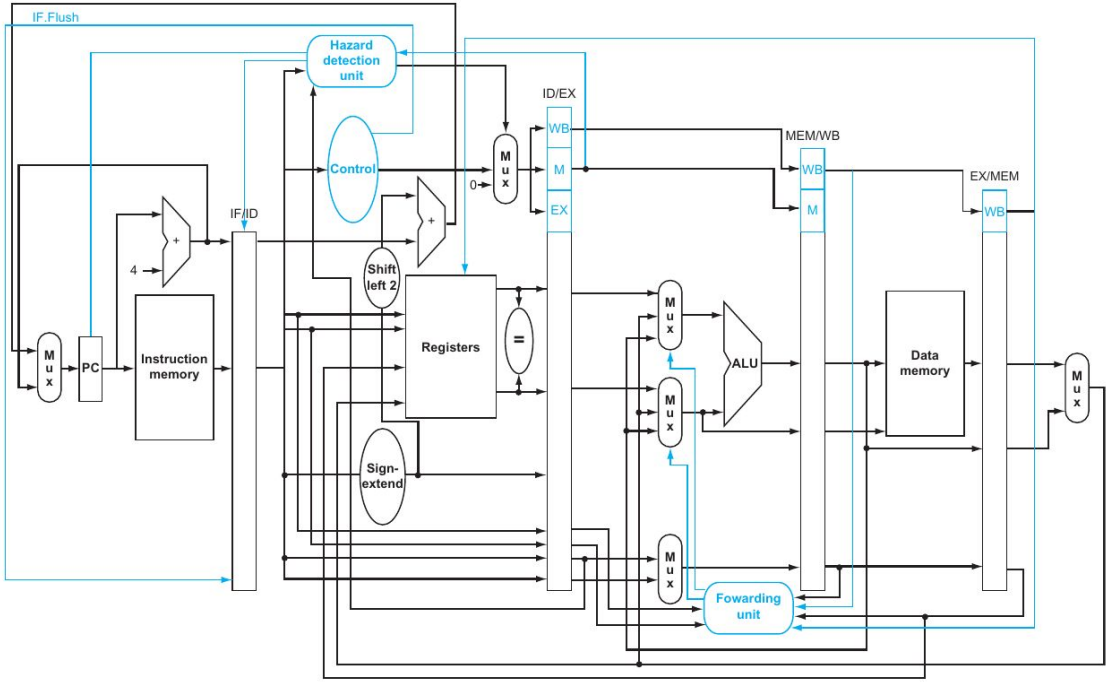
Redução do custo de uma previsão incorreta.

Agora no máximo uma bolha (nop) é necessária devido ao hazard de controle.

Criamos novos hazards de dados!

Por quê? Onde? **Como?**

Para cada problema resolvido, criamos dois novos!



Realizar a comparação no estágio ID

Os operandos do beq podem estar sendo calculados em algum estágio do pipeline.

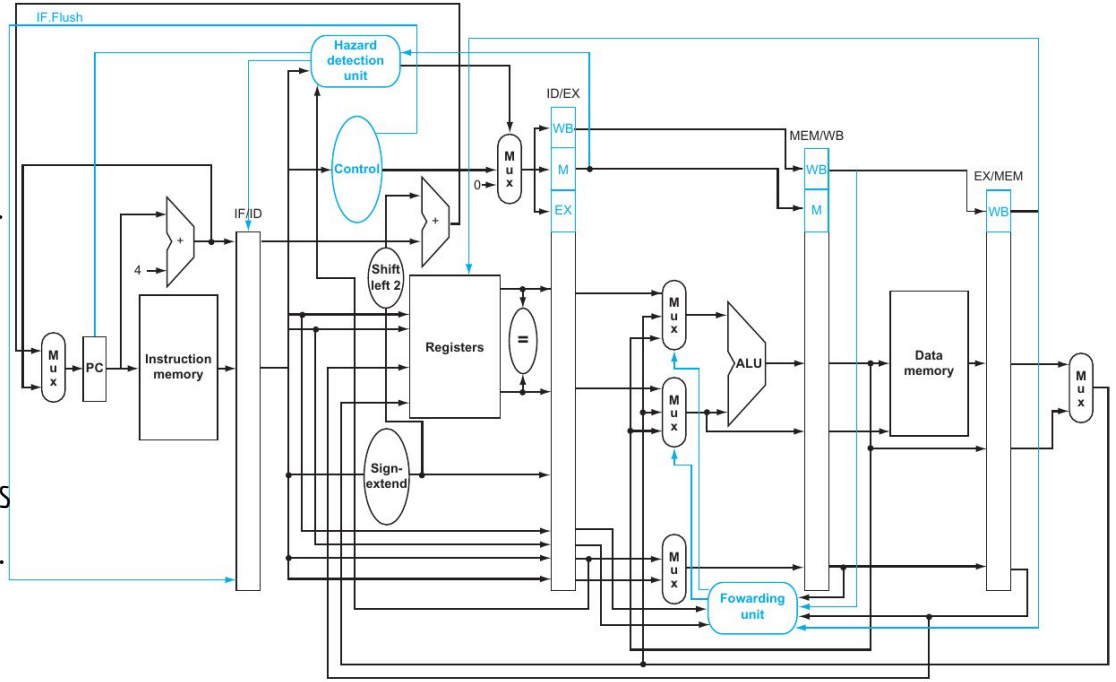
Necessária lógica extra de forwarding para o estágio ID.

Detectar hazards de dados sem solução, para inserir nops.

Ex.: um lw antes de um beq, sendo que o beq precisa do dado do lw.

Mais complexidade na unidade de detecção de hazards.

Vamos nos contentar em saber que esses novos problemas existem, mas não vamos colocar o hardware para resolver.



Relembrando...

O custo de uma previsão incorreta pode ser excessivamente alto em uma CPU de pipeline profundo.

Várias instruções podem ser descartadas!

Um pipeline profundo é ideal se conseguimos mantê-lo cheio.

Mas é complexo, e stalls custam caro.

Difícil manter o pipeline sempre cheio.

Microprocessador	Ano	Clock	Estágios Pipeline	Tamanho Despacho	Fora de ordem?	CPUs por Chip	Potência
486	1989	0,025 GHz	5	1	Não	1	5 W
Pentium	1993	0,066 GHz	5	2	Não	1	10 W
Pentium Pro	1997	0,2 GHz	10	3	Sim	1	29 W
Pentium 4 Willamette	2001	2 GHz	22	3	Sim	1	75 W
Pentium 4 Prescott	2004	3,6 GHz	31	3	Sim	1	103 W
Intel Core	2006	3 GHz	14	4	Sim	2	75 W
Core i7 Nehalem	2008	3,6 GHz	14	4	Sim	2-4	87 W
Core Westmere	2010	3,73 GHz	14	4	Sim	6	130 W
Core i7 Ivy Bridge	2012	3,4 GHz	14	4	Sim	6	130 W
Core Broadwell	2014	3,7 GHz	14	4	Sim	10	140 W
Core i9 Skylake	2016	3,1 GHz	14	4	Sim	14	165 W
Intel Ice Lake	2018	4,2 GHz	14	4	Sim	16	185 W

Melhorando a previsão

As técnicas discutidas em aulas passadas podem ser implementadas para aumentar a acurácia da previsão.

Melhorando a previsão

As técnicas discutidas em aulas passadas podem ser implementadas para aumentar a acurácia da previsão.

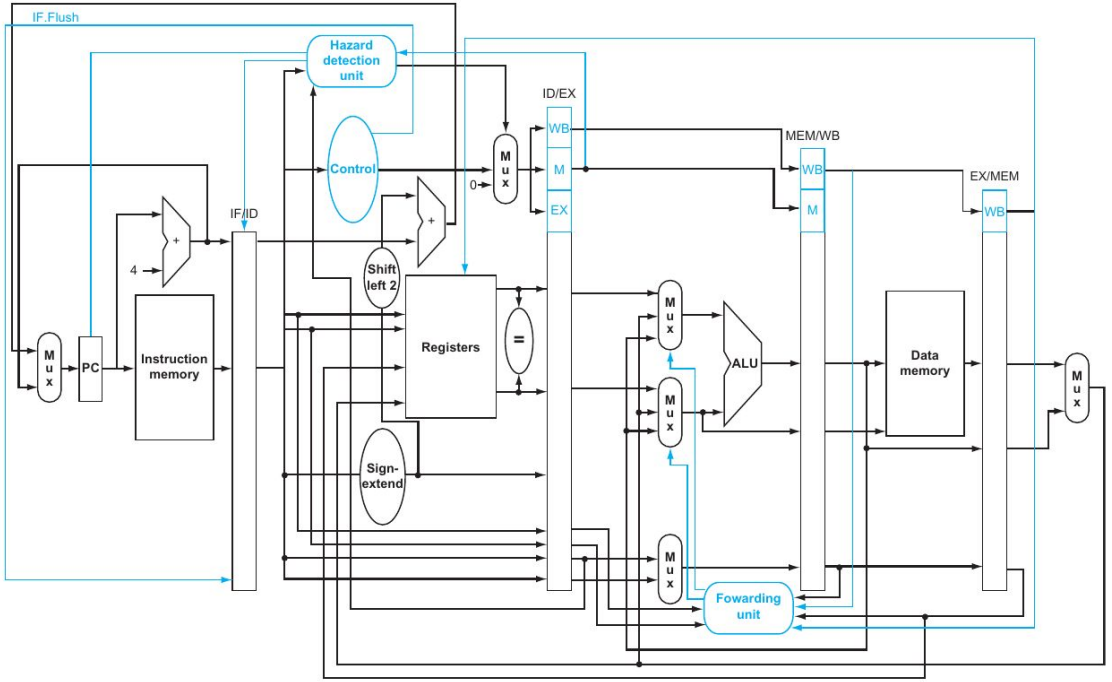
- Buffers de previsão de desvios.
- Delayed Slots.
- Preditores de correlação.
- Preditor de torneio.
- ...

Melhorando a previsão

As técnicas discutidas em aulas passadas podem ser implementadas para aumentar a acurácia da previsão.

- Buffers de previsão de desvios.
- Delayed Slots.
- Preditores de correlação.
- Preditor de torneio.
- ...

Podem ser instalados no **estágio IF** do pipeline.

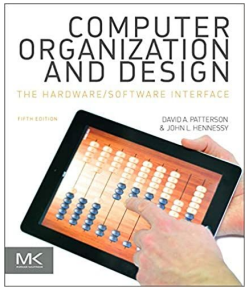


Exercícios

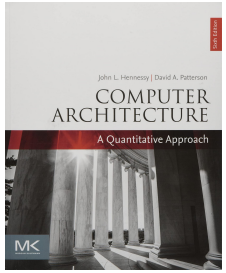
1. Releia os slides, analisando cada ponto de forma detalhada.
2. Quanto ao buffer de previsão de desvios, quais informações precisamos realmente manter em uma memória? Podemos armazenar somente o bit dizendo se o salto foi realizado ou não (segunda coluna – veja em aulas passadas) e não armazenar o endereço da entrada no buffer (primeira coluna)?

Referências

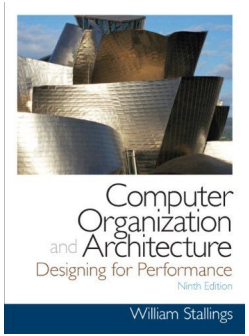
Patterson, Hennessy . Arquitetura e Organização de Computadores: A interface hardware/software. 2014.



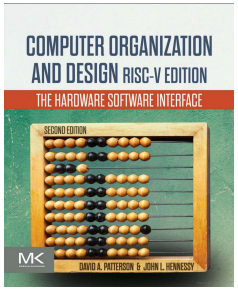
Hennessy, Patterson. Arquitetura de Computadores: uma abordagem quantitativa. 2019.



Stallings, W. Organização de Arquitetura de Computadores. 10a Ed. 2016.



Patterson, Hennessy. Computer Organization and Design RISC-V Edition: The Hardware Software Interface. 2020.



Licença